The apeNEXT project*

The APE Collaboration: F. Bodin[a], Ph. Boucaud[b], N. Cabibbo[c], F. Di Carlo[c], R. De Pietri[d], F. Di Renzo[d], W. Errico[e], H. Kaldass[f], A. Lonardo[c], S. de Luca[c], J. Micheli[b], V. Morenas[g] O. Pene[b], D. Pleiter[h], N. Paschedag[f], F. Rapuano[i], D. Rossetti[c], L. Sartori[e], F. Schifano[e], H. Simma[f], R. Tripiccione[j], P. Vicini[c]

[a]IRISA/INRIA, Campus Université de Beaulieu, Rennes, France

[b]LPT, University of Paris Sud, Orsay, France

[c]INFN, Sezione di Roma, Italy

[d]Physics Department, University of Parma and INFN, Gruppo Collegato di Parma, Italy

[e]INFN, Sezione di Pisa, Italy

[f]DESY Zeuthen, Germany

[g]LPC, Université Blaise Pascal and IN2P3, Clermont, France

[h]NIC/DESY Zeuthen, Germany

[i]Physics Department, University of Milano-Bicocca, Italy

[j]Physics Department, University of Ferrara, Italy

In this talk I report on the status of the apeNEXT project. apeNEXT is the last of a family of parallel computers designed, in a research environment, to provide multi-teraflops computing power to scientists involved in heavy numerical simulations. The architecture and the custom chip are optimized for Lattice QCD (LQCD) calculations but the favourable price performance ratio and the good efficiency for other kind of calculations make it a quite interesting tool for a large class of scientific problems.

1. The APE project

The APE project started in 1984 with the design and manufacture of a 1 GFlops parallel computer for LQCD. The original group had only theoretical and a few experimental physicists involved [1]. During the years students in physics and computer science have joined the collaboration to build the large community that today designs and uses APE computers. A further enlargement of the collaboration came by the end of the APEmille project when the DESY and the Orsay group joined in. Table 1 summarizes the main features of all computers of the APE family.

As one can see in table 1, all APE machines before apeNEXT are based on a SIMD architecture. Since APEmille, however, the possibility of local addressing, to increase the ease of programming in certain classes of problems, has been introduced. The term "flexible" in the topology row refers to the possibility of performing hardware-controlled next-to-nearest-neighbour communication. The number of registers, the clock speed and the word size have increased following the evolution of technology. An important step, taken during the APE100 initial phase, was the custom design of the VLSI chips. At that time, in fact, a certain number of software packages for the schematic capture, the simulation and the VLSI synthesis became available to ordinary users outside the design centers of large companies. This allowed the collaboration to develop

*Talk given by F. Rapuano at the Lattice Conference 2004, Fermilab (IL), USA.

Table 1
The family of APE processors. The year in parenthesis is the time when the project was concluded. Physics runs in general have started quite earlier on prototypes or small scale machines.

	APE(1988)[1]	APE100(1993)[2]	APEmille(1999)[3]	apeNEXT(2004)[4]
Architecture	SISAMD	SISAMD	SIMAMD	SPMD
Number of nodes	16	2048	2048	4096
Topology	flexible 1D	rigid 3D	flexible 3D	flexible 3D
Memory	256 MB	8 GB	64 GB	1 TB
Registers (Word Size)	64(32)	128(32)	512(32)	512(64)
Clock speed	8 MHz	25 MHz	66 MHz	200 MHz
Peak speed	1 GFlops	100 GFlops	1 TFlops	7 TFlops

the main building blocks of APE machines with all and only those devices relevant for the functionalities needed for our purposes. In particular one of the main characteristics of all APE floating point processors is to perform, at each clock cycle a "normal" operation $a \times b + c$, where a,b and c are complex numbers. This operation is the most performed operation in a LQCD program so it is crucial that the processor perform this operation as fast as possible to obtain high efficiencies in the application programs. Furthermore a custom design allows to keep power consumption as low as possible which is a relevant point when one plans to have thousands of such components in a parallel machine. A final comment can be made regarding clock speeds. One may notice that in all projects the clock frequency is quite low while gaining on floating point speed from parallelism of operations. A low clock frequency allows to avoid sophisticated technical solution for boards manufacturing and reduces the possibility of errors due to tight timings in control and data signals thus improving operating reliability. Without going into the details of each machine one can summarize the main rules that have always been followed in the design of APE computers:

- The computer should be very efficient for LQCD calculations (and in fact efficiencies up to 65% have been reached in the Dirac operator kernel calculation) but reasonably efficient for other fields.
- A large number of registers for efficient code optimization with no need for difficult to manage cache memories.
- A microcoded architecture with a very long instruction word (VLIW) to have all devices under program control at each clock cycle.
- Reliable and safe hardware solutions.
- A large effort in the system software design to give the user high quality programming and optimization tools.

2. apeNEXT architecture

apeNEXT [4] is the last of the processors designed by the APE group. The goal of the project is to reach multi-TFlops performances needed for state-of-the-art LQCD simulations with fermion loops. The project started in the year 2000, the general ideas were presented at the Bangalore Lattice Conference. It has been concluded by the first quarter of this year with the successful test of all components in a 16 then enlarged to 256 nodes prototype. A large mass production will start by October this year. apeNEXT has an important technical innovations with respect to the previous generations of APE machines: one custom VLSI chip integrates all main functionalities, including network devices. Fig. 1 shows the layout of this chip called J&T.

This $(1.5cm)^2$ chip performs, with 64 bits accuracy, all operations that were performed at 32 bit in the 30x50 cm^2 board shown in fig 2. Furthermore J&T contains 7 bidirectional 200

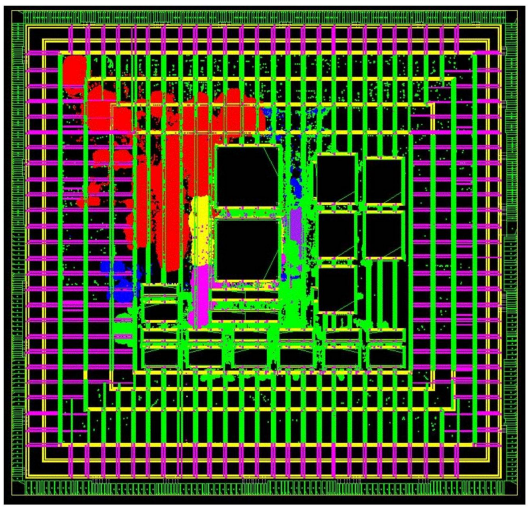

Figure 1. The dye of the apeNEXT processor, J&T.

MBytes/sec serial link interfaces. Six of them are used for nearest neighbour communications in three dimensions while the 7th is used for high speed I/O communications with a front-end PC via a PCI interface board. Fig 3 shows the block diagram of J&T. A slow serial I2C (an industry standard) link is also present on J&T. It is used, as discussed in the following, to allow the front-end to access all nodes at any time via the same PCI interface as the 7th link. Another new feature of apeNEXT compared to previous APE machines is that nodes run asynchronously making apeNEXT a Single Program Multiple Data parallel computer. A resynchronization of the whole machine is automatically performed at every remote I/O operation.

Figures 4 and 5 show the apeNEXT architecture. Nodes are arranged in a 3-dimensional mesh, with first neighbour communications. Each node performs a "normal" operation at 200 MHz for a peak speed of 1.6 GFlops. A 128 bit channel interface supports up to 1 GByte DDR dynamic (single error corrected double error detected) memory with a transfer bandwidth of 3.2GByte/sec and a latency of 15 cycles. Data and instruction words share the same memory.

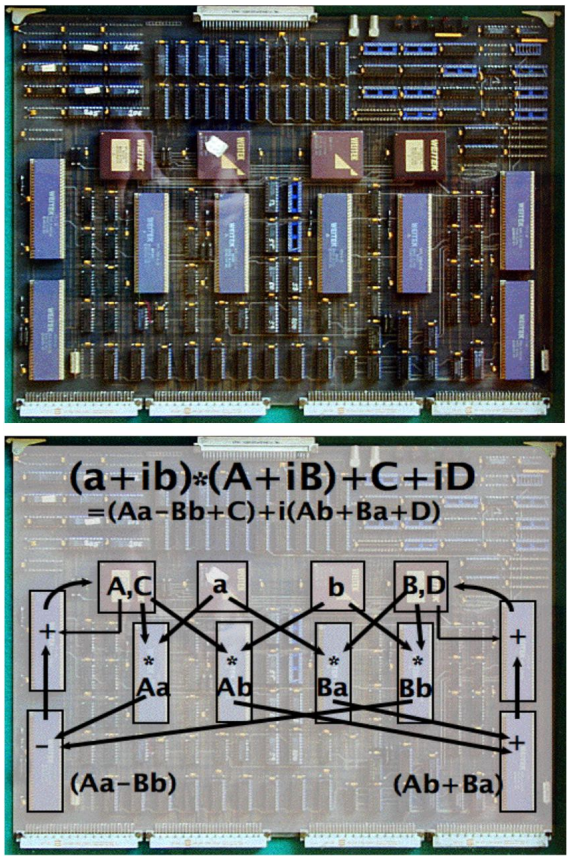

Figure 2. The first APE floating point board. The function of each of the commercial chips in the board is shown in the lower picture.

Code compression and a 4 KWord instruction prefetch buffer keep data and code access conflict possibility to a minimum. Code decompression is performed by specialized hardware.

Remote I/O is performed by the serial links with a bandwidth of 200 MByte/sec (protocol overheads reduce this speed by less than 10%) with a very low latency of 20 cycles. Remote data transfers of each 128 bit block are error protected by a Cyclic Redundancy Check (CRC). Both the local and the remote bandwidths are well balanced for typical LQCD calculations. For further efficiency improvement, however, each link has a data pre-fetch queue. Local or remote data

can be temporarily stored in the pre-fetch queue and then, with zero latency, moved into the register file. This allows the overlap of network and arithmetic operations and, only in the case that a needed data word is not yet available in the data queue, the processor will wait until data are ready.

An apeNEXT board houses 16 nodes each housed with its local memory in a piggy-back module. Very little extra spare logic is needed in the board. 16 boards are housed in a crate in which a backplane supports the remote communications in two of the three dimensions and the delivery of all control signals. Third dimension communications go via front panel inter-crate cables. Nodes topology ranges then from $4 \times 2 \times 2$ of a single board to $4 \times 8 \times 8$ of a single crate to $(8 \times n) \times 8 \times 8$ of large machines where double crates are stacked in the x-direction.

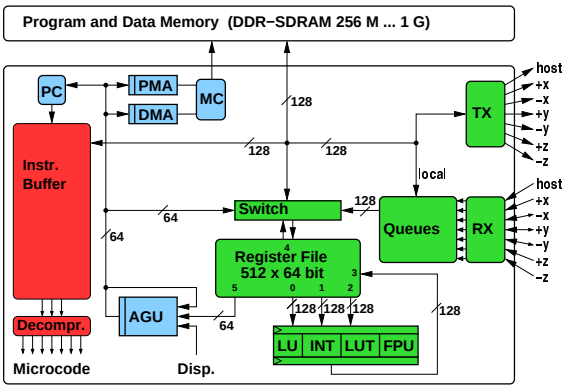

Figure 3. Block diagram of the apeNEXT processor.

3. apeNEXT software

As already stated, a large effort has always been devoted, in the APE collaboration, to the development of software tools so that the coding of scientific problems could be as simple and smooth as possible for the user. The APE software group has developed, since the APE100

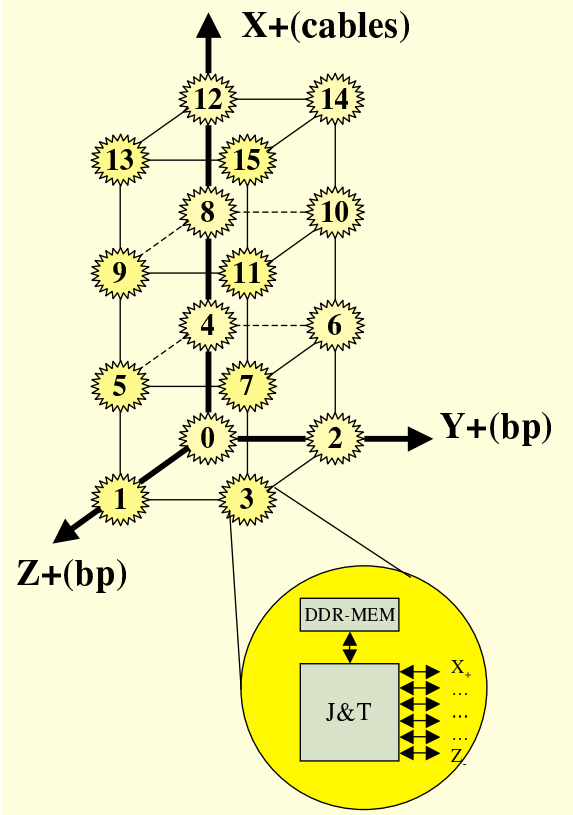

Figure 4. The apeNEXT architecture I.

project, a FORTRAN-like language, TAO, based on a dynamical grammar. The possibility of an easy definition of objects and operator overloading has made TAO a very important tool for the development of LQCD programs that rely on complex data structures (spinor, SU3 matrices, fermion propagators etc.). A very large number of source code lines have been written in TAO by various european groups who have been using APE machines. Backward compatibility has been ensured for apeNEXT users. Very little or no changes in the source code and a recompilation will be needed to run existing TAO programs. For a wider usage a C compiler has been developed for apeNEXT, based of the public domain "lcc" compiler [5] with the (few) necessary extensions needed for a parallel architecture. Figure 6 shows

Table 2
Linear algebra benchmarks

	maximum	assembler	C	C + sofan
vnorm	50%	37%	31%	34%
zdotc	50%	41%	28%	40%

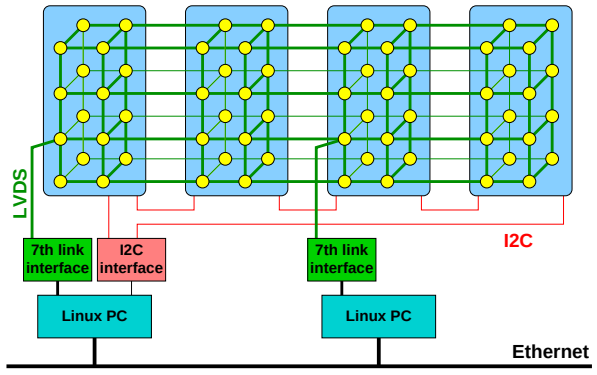

Figure 5. The apeNEXT architecture II.

the structure of the compilation software. Both TAO and C source codes are compiled into a high level assembly code in a first step. Assembly code is also available to the user but, as it will be clear in the following, it will not be in general necessary to obtain best performances. A crucial step in the compiling process is performed, in fact, by the optimization program "sofan". This package, based on the "SALTO" optimization toolkit developed at INRISA, Rennes, takes care of a number of important steps for the optimization of the executable code: generating "normal" operations, removing dead code, eliminating unnecessary register moves, optimizing address generation etc..

Finally the "shaker" perform a last optimization step by scheduling instructions as early as possible, allocates registers and generates compressed executable code. At this point the code can be passed to a VHDL functional simulator for performance analysis or executed after a linker procedure.

Table 2 shows the importance of the optimization step on two typical linear algebra calculations, the norm of a vector and the product of two vectors. In this table the maximum performance is the theoretical performance ignoring the floating point pipeline latency and all loop overheads. It is interesting that once that "sofan" is used the high level C code performs essentially like the highly optimized assembly code. The most important computing kernel in LQCD is the application of the Wilson-Dirac (i.e. the discrete QCD covariant derivative with the Wilson term) on a spinor. In this case, even on a local lattice size of $2^3 \times 16$, in which all sites are on a boundary and so data transfer is always remote, one obtains a sustained performance of 55% and only 4% of processor wait cycles. This last figure shows that one has almost full overlap between arithmetic operations and network activities. This result has been obtained with very simple high level optimization tricks like keeping gluon fields local, 2 sites ahead pre-fetch and some loop unrolling. Even higher performances are obtained on operations like the product of SU(3) matrices, in which the number of floating point operations per memory access is higher, where one get efficiencies as high as 65%.

The operating system of apeNEXT is distributed on the different layers of apeNEXT. I/O requests between nodes are managed by system routines executed on the computing nodes. An I/O operation to or from the front-end PC for data backup or restore will go through the 7th link and the PCI interface. In this case a daemon running on the front-end will detect the I/O request on the fast link and service it without halting the nodes. I/O operations on the 7th links, in fact, will look to the computing nodes as an ordinary inter-node data transfer. A daemon on the front-end via the I2C link, polling on special registers, will detect program halt or exceptions. I2C is also used by the system routines on the front-end for bootstrap and system initialization.

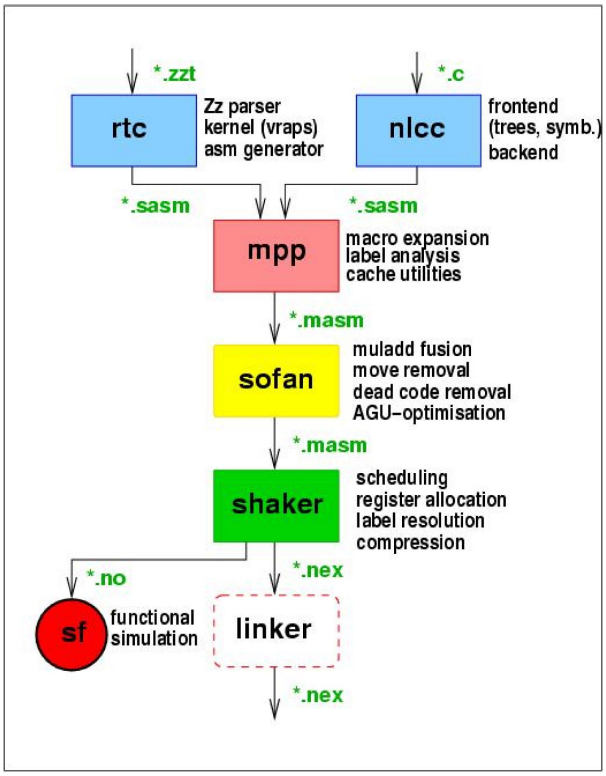

Figure 6. The apeNEXT software.

4. Conclusions and future perspectives

Like all previous APE machine apeNEXT too has a very good price-performance ratio. Total development cost has been of about 1700 KEuro of which 1050 have been spent in the VLSI design and manufacturing and 550 have been spent for all other hardware parts: boards, cabinets, spare components etc. This leads to a prototype production cost of about $0.6-0.7$ Euro/MFlops. Like APEmille, apeNEXT will be commercially available. A large scale production cost of $0.5-0.6$ Euro/MFlops is expected making apeNEXT one of the most cost effective parallel computer available. As apeNEXT prototypes have passed all tests early this year a large installation for a total of about 10 TFlops has been funded by INFN in Italy. These machines will be installed at the University of Rome "La Sapienza". The german groups (DESY and Bielefeld University) and the french group are still in the process of contracting with their funding agencies.

As far as scientific plans are concerned, of course LQCD is the main field of use foreseen for apeNEXT but, as it happened for previous machines, it will surely be used for other fields like turbulence, complex systems and there are plans to implements programs for Quantitative Biology a discipline that has seen a tremendous increase of interest in the last two to three years as one can see on the QBIO archive at arXiv.

Plans for a future architecture development (apeNEXT^2?) are more fuzzy. For sure an R&D activity will continue in the research agencies involved but what a future project might be, an intermediate $2-4$ times performance machine or a 100 TFlops european project on a longer time scale, is not clear yet.

REFERENCES

1. M. Albanese et al. (APE Collaboration):
The APE Computer: an array Processor for Lattice QCD.
Computer Physics Communications 45, 345, (1987).
2. N.Avico et al. (APE Collaboration):
From APE to APE100: from 1 to 100 GigaFlops in lattice simulations.
Comp. Phys. Comm. 57,285,1989.
C. Battista et al. (Ape Collaboration):
The APE100 computer: The architecture.
Intl. Journal of High Speed Computing, 5 (1993) 637.
Also in: Field Theory, Disorder and Simulations (G. Parisi editor),
World Scientific (Singapore) 1992, 488.
A. Bartoloni et al.(Ape Collaboration):
The software of the APE-100 Computer.
Intl. Journal of Modern Physics C, 4 (1993) 955.
3. F. Aglietti, et al. (Ape Collaboration):
An Overview of the APEmille Parallel Computer.
Nucl. Instr. and Methods A389 (1997) 56.
4. R. Alfieri et al. (APE-collaboration), "apeNEXT: A Multi-Tflops LQCD Comput-

ing Project", 2001 [arXiv:hep-lat/0102011].
5. C.W. Fraser, D.R. Hanson, D. Hansen, "A Retargetable C Compiler: Design and Implementation", 1995.